

Inteligencia Artificial

A milk collection problem

Claudia Hazard Valdés

August 21, 2018

Resumen

Muchas empresas necesitan reparto o acopio de materiales en distintos puntos teniendo una cierta cantidad de vehículos disponibles. Para realizar el acopio de leche se ha definido el Milk Collection Problem With Blending, el cual consiste en un conjunto de granjas productoras de leche que deben ser recogidas por camiones sin exceder sus capacidades, pudiendo realizar mezclas de la leche disminuyendo su calidad, buscando maximizar la ganancia obtenida. Debido a la aplicabilidad del problema a distintas empresas productoras de leche, se ha realizado un estudio de las de las soluciones implementadas para el MCWB. Este artículo consiste en definir el problema, además de como ha evolucionado en el tiempo, mostrar distintos métodos que se han implementado para resolverlo, proponer una representación, implementación de dos algoritmos y mostrar los resultados de estos.

Keywords: Milk collection.

1 Introducción

Este trabajo se centra en un estudio del problema Milk collection problem With Blending para un conjunto de granjas productoras y un conjunto de camiones, que consiste en tener granjas productoras cada una con su propia cantidad y calidad de leche, y una cantidad de camiones determinado cada uno con su propia capacidad, los cuales deben realizar el acopio de la leche de cada granja sin exceder su capacidad, pudiendo realizar mezcla dentro de los tanques, donde la leche final será la de menor calidad, determinando las rutas para cada camión que generen la mayor ganancia.

Este problema es aplicable a las empresas productoras de leche, tal como se realizó en el artículo de Paredes-Belmar et al. [11] para empresas Chilenas que utilizan camiones con un solo compartimiento y permiten realizar mezclas de leche en ellos, que buscan obtener la mayor ganancia posible que conlleva este procedimiento.

La motivación para la resolución de este problema se basa en la magnitud que abarca la solución del MCWB ya que para todas las empresas es necesario obtener las mayores ganancias posibles y las rutas realizadas por los camiones son realizadas recurrentemente por lo que genera una ganancia aun mayor a nivel temporal, junto con la dificultad para encontrar los mejores recorridos, pues con una amplia cantidad de nodos, no es factible de forma visual definir buenas rutas. Es por esto que resulta importante para las empresas obtener rutas óptimas.

Para interiorizar el problema, se definirá en que consiste ,sus variables,objetivos, problemas relacionados y variaciones.Luego se generará un modelo matemático explicando cada una de las partes, mostrando la función a minimizar, variables y restricciones. Después se presentará un Estado del Arte del problema. Realizando una contextualización, describiendo la información que existe a los estudios realizados, tendencias para resolver el problema, analizando, métodos que han sido utilizados para resolverlo y sus resultados.Posteriormente se mostrará una propuesta realizada, junto con los experimentos y resultados para algunas de las instancias utilizadas en el artículo de Paredes-Belmar et al. [11]. Finalmente se concluirá respecto a lo mas relevante del estudio realizado, sobre las técnicas sus percepciones, limitaciones y el posible trabajo a futuro.

2 Descripción del problema

En una gran cantidad de países se realiza producción de leche en distintas fabricas. Por esto resulta necesario realizar el traslado a través de camiones para reunirla en la planta. El Milk collection problem es un problema de transporte (VRP) con pickup and delivery services (PDP), lo que lo hace NP-difícil [17].Consiste en utilizar camiones que tienen tanques con una capacidad máxima para llevar la leche desde distintas fabricas a la planta existiendo distintos tipos de calidades de leche buscando optimizar las ganancias generando rutas para cada uno de ellos, es decir, el costo de transporte de los camiones.

El problema puede verse como un grafo, donde los nodos corresponden a las distintas fabricas productoras de leche junto con la cantidad de leche de un tipo de calidad producida en dicha fabrica y los arcos el costo de llevar un camión de un nodo a otro. Existen distintas versiones del problema entre estos el solo tener una calidad de leche [9], tener varias calidades de leche en las que se puede llevar solo una por camión o mas de una en distintos tanques del mismo camión [15], como también el caso mas reciente de tener la opción de mezclar en un mismo tanque la leche cambiando su calidad [11].

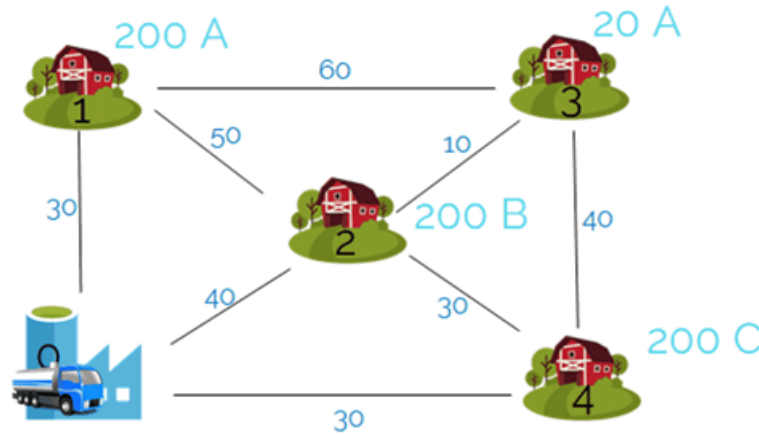


Figura 1: Diagrama del Milk collection problem.

Se puede observar en la Figura 1, un conjunto de granjas (1, 2, 3, 4) y una planta procesadora (0), donde un conjunto de camiones deberá recorrer de forma óptima cada granja, acaparando toda la leche que esta contenga, además se aprecia que cada arco tiene un costo asociado a la

distancia que el camión deberá recorrer. Procediendo a explicar los movimientos posibles, se tiene que el camión siempre partirá y terminará su recorrido en 0. Para explicar como funciona el blending se dará un ejemplo, al recorrer los nodos de la forma $0 \rightarrow 1 \rightarrow 2 \rightarrow 0$ se obtendrá 400 de leche tipo B, pues se realizó blending entre la granja 1 y 2 quedando la leche de menor calidad, mientras que al realizar $0 \rightarrow 1 \rightarrow 3 \rightarrow 0$ se obtendrá 220 de leche tipo A pues el camión no recogió de otro tipo.

Por lo actual y poco realizado del problema Milk Collection with Blending (MCwB) se trabajará con esta versión, además de mostrar como se han abordado distintas versiones del problema.

2.1 Modelo Matemático

El modelo matemático presentado, esta basado en el realizado por Paredes et al. [11], siguiendo con lo comentado en la descripción del problema se procede a definir formalmente los parámetros y variables.

- Parámetros:

- q_{ti} : Leche de tipo t producida en la fabrica(nodo) i .
- c_{kij} : Costo de que el camión k viaje de i a j .
- α^t : Ingresos por unidad de leche de calidad t .
- p_t : Cantidad mínima que la planta debe cumplir de leche tipo t .
- Q_k : Capacidad del camión k .

- Variables:

- W_{kt} : Volumen de la leche de tipo t que el camión k lleva a la planta.
- V_{kt} : Volumen de la leche de tipo t que es entregada en la planta y mezclada tiene valor tipo r .
- $x_{kij} = \begin{cases} 1 & \text{si el camión } k \text{ viaja directamente de } i \text{ a } j. \\ 0 & \text{sino} \end{cases}$
- $y_{kti} = \begin{cases} 1 & \text{si el camión } k \text{ carga leche del tipo } t \text{ en la fabrica } i. \\ 0 & \text{sino} \end{cases}$
- $z_{kt} = \begin{cases} 1 & \text{si el camión } k \text{ lleva leche del tipo } t \text{ a la planta.} \\ 0 & \text{sino} \end{cases}$

2.2 Función Objetivo

La función objetivo busca maximizar los ingresos del total de cada calidad de leche en planta (si ha sido mezclada su calidad baja) tomando en cuenta los costos de viaje de cada camión.

$$Z = \text{Max} \sum_{t \in T} \sum_{r \in T} \alpha^r v^{tr} - \sum_{(i,j,k) \in AK} c_{ij}^k x_{ij}^k \quad (1)$$

2.3 Restricciones

Limitar la capacidad de cada camión, para que no recoja más que su capacidad máxima.

$$\sum_{t \in R} \sum_{i \in N; (i,t) \in IT} q_i^t y_i^{kt} \leq Q^k \quad (2)$$

Solo un camión puede recoger la leche de una determinada calidad de una granja.

$$\sum_{k \in K_i} y_i^{kt} = 1, \quad t \in T : (i, t) \in IT \quad (3)$$

Cada camión puede realizar como máximo un solo viaje, saliendo desde la planta productora.

$$\sum_{j: (0_k j, k) \in AK} x_{0_k j}^k \leq 1, \quad \forall k \in K \quad (4)$$

Si un camión llega a una granja y luego sigue su recorrido, entonces el camión debe salir desde la misma granja.

$$\sum_{i: (i, j, k) \in AK} x_{ij}^k = \sum_{h: (j, h, k) \in AK} x_{jh}^k, \quad \forall k \in K_j, \quad j \in N_0 \quad (5)$$

Si un camión se detiene en una granja, entonces debe recolectar la leche de dicha granja.

$$\sum_{p: (p, i, k) \in AK} x_{pi}^k = y_i^{kt}, \quad \forall k \in K_i, \quad i \in N, \quad t \in T : (i, t) \in IT \quad (6)$$

Un camión no puede recoger leche de menor calidad que la que lleva si este está en camino a entregar su leche.

$$z^{kt} \leq 1 - \sum_{r \in D^t; r \neq t, (i, r) \in IT} y_i^{kr}, \quad \forall k \in K_i, \quad i \in N, \quad t \in T \quad (7)$$

Un camión no puede llevar más de un tipo de leche cargado.

$$\sum_{t \in T} z^{kt} \leq 1, \quad \forall k \in K \quad (8)$$

Se relaciona la variable continua w^{kt} , o el volumen de la leche t , y la variable binaria z^{kt} , o la asignación de leche de calidad t al camión k .

$$w^{kt} \leq z^{kt} Q^k, \quad \forall k \in K, \quad t \in T \quad (9)$$

Toda la leche de calidad t que se entregó en la planta productora debe ser menor o igual a la cantidad de leche de calidad t que se extrajo de las granjas.

$$w^{kt} \leq \sum_{r: t \in D^r} \sum_{h \in N^r} q_h^r y_h^{kr}, \quad \forall k \in K, \quad t \in T \quad (10)$$

Toda la leche que se recolecto debe llegar a la planta.

$$\sum_{k \in K} \sum_{t \in T} w^{kt} = \sum_{(i, t) \in IT} q_i^t \quad (11)$$

Para balancear las cantidades de cada tipo de leche que llega a la planta, y la cantidad de leche que sobra de cada calidad luego de que se mezclan en la planta se utiliza la siguiente restricción.

$$\sum_{r \in D^t} v^{tr} = \sum_{k \in K} w^{kt}, \quad \forall t \in T \quad (12)$$

La cantidad de leche producida debe ser mayor o igual que la cuota exigida para cada tipo de leche.

$$\sum_{t \in T} v^{tr} \geq P^r, \quad \forall r \in D^t \quad (13)$$

Se evitan mezclas de leche prohibidas.

$$y_i^{kt} + y_i^{kr} \leq 1, \forall (t, r) \in PM; (i, t), (j, t) \in IT \quad (14)$$

Se prohíben los subtours que pueden realizar los camiones.

$$\sum_{i \in S} \sum_{j \in S} x_{ij}^k \leq |S| - 1, \forall S \subseteq N, k \in K \quad (15)$$

Se definen los dominios de cada variable.

$$y_i^{kt}, z^{kt} \in \{0, 1\}, \forall i \in N, k \in K_i, t \in T : (i, t) \in IT \quad (16)$$

$$x_{ij}^k \in \{0, 1\}, \forall (i, j, k) \in AK \quad (17)$$

$$w^{kt}, v^{tr} \geq 0, \forall k \in K; t, r \in T, r \in D^t \quad (18)$$

3 Estado del Arte

El TSP (Traveling Salesman Problem) fue formulado como un problema matemático en 1930 y luego se volvió popular en 1950, trata de encontrar un camino para visitar cada ciudad (nodo) exactamente una vez, volviendo finalmente a la ciudad inicial, minimizando el costo del viaje. Es un problema del tipo NP-Hard [17] el cual a través de los años se ha buscado solución con distintos métodos como Nearest Neighborhood Search, Simulated Annealing, Tabu Search, Redes Neuronales, Ant Colony System y Algoritmos Genéticos.

El TSP ha dado lugar a otros problemas como el VRP (Vehicle Routing Problem) primera vez mencionado por Dantzing y Ramser [4], que se basa en un conjunto de nodos y arcos con un peso asociado al igual que un TSP pero existen una cantidad de vehículos con características similares que recorren un camino para satisfacer la demanda de cada nodo y minimizando el coste total de las rutas realizadas. Para solucionarlo se han utilizado heurísticas constructivas las cuales se apegan a reglas pero no mejoran la solución [12], heurísticas de mejora [18], que mejora una solución iterativamente y meta-heurísticas como algoritmos genéticos [19], simulated annealing [1], programación dinámica [16] y algoritmos genéticos con búsqueda local [6] .

A los VRP pueden agregarse distintas condiciones para crear problemas mas específicos como el VRPPD (Pickup and delivery) donde debe recoger y además dejar elementos a medida que se avanza en el recorrido, 2L-CVRP [5] donde se considera una superficie de carga rectangular bidimensional y se restringe para la carga y descarga secuencial, 3L-CVRP [7] lo mismo mencionado anteriormente pero ahora la superficie será tridimensional o McP (Milk collection Problem)[11] el cual es estudiado en esta ocasión.

En el artículo de Butler, Herlihy y Keenan [2] se menciona el uso de software para realizar recolección de leche para facilitar la planificación y rutas de los camiones. Se toma en cuenta la investigación operacional en la recolección de leche, el software entrega una solución visual del problema de enrutamiento minimizando la distancia el cual al comienzo era difícil de usar por gente sin conocimientos informáticos pero se realizó una interfaz para que pueda ser usado mas fácilmente. Para mejorar los resultados se utilizaron datos mas reales de los caminos y también utilizar computadoras mas rápidas. Se menciona que las rutas en la recolección de leche son estáticas no como en otros VRP por lo que los software utilizados han sido creados para otro tipo de industrias y no son capaces de modelar el proceso de recolección de leche. Al agregar

restricciones específicas el tiempo de ejecución aumentó excesivamente además estos programas buscaban remplazar al planificador. Buscando una mejor solución se utiliza DDS (Decision support systems) donde el planificador usa el software como herramienta, el enrutamiento es de las áreas principales de utilización de DDS y comenzó a ser usado en la industria de la leche en 1990. Los sistemas de información geográfica han ayudado en el enrutamiento DSS. Con los datos de DDS se pudo asignar los sectores que debía recorrer cada camión. El algoritmo de agrupamiento utilizado últimamente por los DSS se basa en puntos semilla, siendo el planificador el que elije los iniciales y estos crecen minimizando la distancia a los puntos de recolección, de esta forma el algoritmo encuentra buenos resultados al ser el planificador el que entrega el punto de partida.

Como se mencionó en el artículo el Milk collection Problem [11] es bastante complejo de representar por las variadas restricciones que tienen según las empresas, además del costo en cuanto a tiempo y no muy buena calidad de resultados tomándolo como un VRP convencional. Actualmente se han realizado formas de resolver el McP obteniendo soluciones adecuadas para las empresas. Una de estas es la realizada por Masson, Lahrichi y Rousseau ?? plantean el problema de la recolección de leche en el tiempo, las plantas tienen una demanda por semana y cada fábrica es visitada por una ruta de camión que retira toda la leche. Hay una cantidad de productores y una cantidad de plantas. Se minimiza la distancia que se cubre cada día para recolectar y llevar la leche. Los productores se visitan en la misma secuencia por una ruta pero las plantas visitadas pueden cambiar. Utiliza Adaptive Large Neighborhood Search (ALNS). Para realizar la solución primero asume que todas las plantas tienen la misma demanda y una ruta solo puede visitar una planta utilizando ALNS. Luego se generan variaciones de las rutas y se seleccionan las mejores para cada planta. ALNS destruye utilizando el método de ruleta y repara de manera iterativa para construir soluciones guardando siempre la mejor solución y moviéndose solo en caso que se cumpla el criterio de aceptación. Utiliza 3 operadores para destruir Random removal, Worst removal y History removal la que consiste en que luego de un tiempo elije los que tengan un bajo puntaje para un nodo visitado. Los operadores de reparación utilizados son mejor inserción y K-regret el cual depende de cada iteración para seleccionar el mejor nodo a insertar. El código realizado está hecho en C++, utilizó las instancias de Cordeau, Gendreau, y Laporte (1997) [3], los mejores resultados se obtuvieron con 5 clusters siendo mejor que en la mayoría de casos que los casos de DTP, pero con más clusters no se obtuvieron tan buenos resultados.

Otra variación del problema presentado por Sethanan y Pitakaso [15] que resuelven el problema en donde cada camión tiene 3 compartimientos, pero la leche de distintos centros o distintos tipos no puede ser llevada en el mismo compartimiento. En muchos casos cada nodo debe ser visitado más de una vez por distintos vehículos. Se considera el costo de viaje y de limpieza de los camiones por lo que se minimiza los costos utilizando múltiples compartimientos en un camión y un máximo de duración de una ruta.

Para resolver el problema utiliza un algoritmo evolutivo, genera una solución inicial se crean vectores de dimensión es igual a la cantidad de nodos con valores aleatorios posteriormente realiza una mutación combinando 3 vectores aleatorios luego se combinan el vector mutado con el vector objetivo. Se implementó el algoritmo en C++ el criterio de término son 10 minutos. Un vecindario muy grande no entrega una buena solución. Se deja el número de vecindarios como $NP/6$ que corresponde al número de clientes y el algoritmo produjo buenos resultados en comparación al tradicional DE.

Otra de las variaciones propuestas es MCPWB (Milk collection Problem With Blending) [11] el cual resuelve el problema donde un camión tiene un solo compartimiento en donde puede mezclar la leche pero esta descenderá en su calidad, maximizando los ingresos, tomando en cuenta la demanda que se debe satisfacer de cada tipo de leche.

Para resolver el problema utiliza un algoritmo evolutivo, genera una solución inicial se crean vectores de dimensión igual a la cantidad de nodos con valores aleatorios, luego realiza una mutación combinando 3 vectores aleatorios.

Se utilizaron 9 instancias de TSPLib set (Reinelt, 1991) y 22 instancias pertenecientes al trabajo de Augerat et al. (1995), 6 del trabajo de Taillard (1999) y 3 de Fisher (1994), en total 40 instancias de entre 23 y 101 nodos, todas las instancias con 3 tipos de calidades de leche. De los resultados obtenidos mas de la mitad de los casos tenia mezcla de las calidades de leche. Se utilizan heurísticas de dividir el problema por secciones para las instancias mas grandes. Se realizan pruebas utilizando mezcla y sin utilizarlo, logrando mejores resultados al mezclarla disminuyendo la cantidad de camiones, además se prueba el no visitar fabricas que tengan un costo muy alto, mejorando las soluciones en algunas instancias.

Para el mismo problema nombrado anteriormente, se realiza una variación al poder mezclar la leche y además agregar puntos de recolección [10], donde algunos nodos pueden entregar en ciertos puntos una cantidad de leche. Para instancias pequeñas utiliza el algoritmo Branch and Cut y para instancias mas grandes utiliza una heurística de tres etapas en las que se asignan nodos candidatos como punto de recolección, se utiliza Ant Colony System (ACS) para generar una ruta y luego se elije la mejor solución. Se utiliza un caso de estudio real que no se encuentra disponible para uso. En cuanto a resultados el algoritmo resuelve el problema con menores costos de transporte, además de mayores ganancias en instancias pequeñas de 23 nodos, pero utilizando un tiempo mucho mayor que los otros algoritmos como con CVRP de 50 [s] vs MCBP con 4737 [s]. Mientras que para instancias grandes de 500 nodos, también obtuvo mejores ganancias que los demás algoritmos como también un tiempo mayor, pero utilizando su heurística HP propuesta los tiempos disminuyeron (aun sin ser menores que los otros métodos) y las ganancias aumentaron aún más.

4 Propuesta

4.1 Representación

Ya que cada camión deberá tener su propia ruta, la representación usada son arreglos que contienen las rutas de cada camión partiendo y terminando en el nodo 1 que será la planta procesadora como se ve en la Figura 2.

C1:	1	2	5	8	12	1
C2:	1	3	4	10	1	
C3:	1	11	9	6	7	1

Figura 2: Representación utilizada en el modelo propuesto.

4.2 Función de evaluación

La función de evaluación utilizada, corresponderá a la cantidad de leche de cada granja que visitó el camión por el valor de la leche final que quedo en el contenedor, en caso de existir

mezcla reducirá su valor. Además, se toma en cuenta la distancia total que deberá recorrer cada camión.

$$Max \sum_{c \in C} \sum_{g \in R} Q_g \dot{V}_c - \sum_{c \in C} \sum_{g \in R}^{R-1} C_{g(g+1)}$$

Se quiere encontrar la solución con el mayor valor posible, es decir, ganancias en la función de evaluación.

4.3 Acercamiento general y estructura

Dado que se trata de un grafo y se deben realizar rutas buscando menores costos de viaje junto con mayores ganancias de leche, se decidió iniciar utilizando una heurística llamada K-GWO [8] para inicializar cada integrante de una población de lobos, basada inicialmente en el algoritmo WSA (Wolf search algorithm) por R. Tang et al. [14], junto con el análisis de los mismos autores en [13] con la modificación AWSA.

El algoritmo K-GWO [8] busca clusterizar los nodos, en este caso se clusterizan en grupos según la cantidad de camiones, de esta forma cada cluster será el camino de un camión y así éste buscará partir en caminos con nodos cercanos, es decir, menor costo de viaje. El algoritmo se muestra a continuación:

Algorithm 1 Algoritmo de inicialización

```

1: procedure K-GWO(Grafo)
2:    $x_i \in \textit{Grafo}$ 
3:   for cada camión do
4:     Agregar NodoAleatorio a C ▷ Agregar nodo no repetido a cluster
5:   for nodo en C do
6:     calcular distancia entre cada nodo  $x_i$  y todos los centroides
7:     while  $x_i$  no este asignado do
8:       if la capacidad no se supera then
9:         Asignar  $x_i$  al centroide mas cercano
10:      else
11:        Asignar  $x_i$  al centroide con capacidad menos superada y
12:        que genera menor diferencia de tipos de leche.
13:   return  $C$  ▷ Retornar el conjunto de clusters correspondiente para cada camión.

```

A diferencia de el algoritmo presentado por Korayem et. al [8], se le agrega además una penalización por diferencia de tipos de leche para adecuarlo al Milk collection with blending, de esta forma además de priorizar la capacidad se ayuda a priorizar que se cumplan las cuotas mínimas de leche.

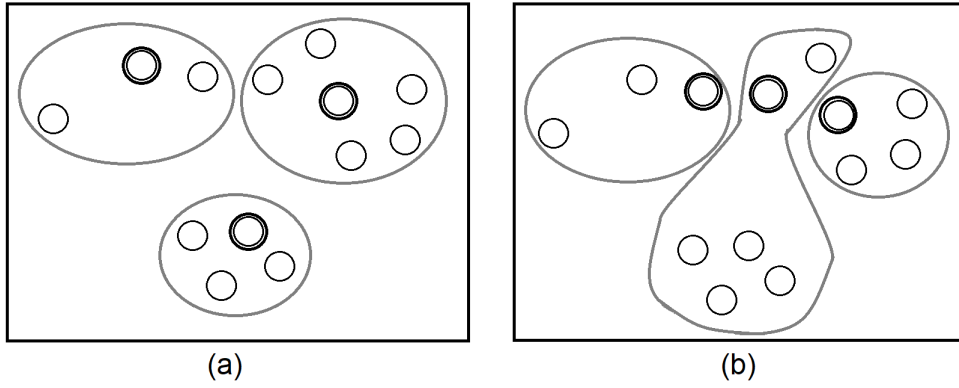


Figura 3: Representación utilizada en el modelo propuesto.

En la Figura 3 se observa en la parte (a) como quedaría un cluster luego de ser seleccionados 3 nodos aleatorios de forma conveniente, mientras que en la parte (b) se observa que al ser escogidos 3 nodos que se encuentran demasiado cercanos no logran abarcar grupos cerrados, sino que, suelen tener mayores distancias. Además, para el milk collection problem, al ser necesaria una cuota mínima de leche, aunque los clusters sean los que generen las mejores distancias puede que no cumplan con la cuota mínima de leche de cada tipo necesaria.

Para encontrar la mejor solución posible se proponen dos algoritmos bastante similares, el primero es denominado MWSA (Milk wolf search algorithm), basado inicialmente en el WSA (Wolf Search Algorithm) [13] pero mezcla distintas técnicas de búsqueda local y reparación. El segundo es nombrado RMWSA (Restrictive Milk wolf search algorithm), es similar a MWSA pero no permite movimientos a soluciones infactibles e intenta realizar mas reparaciones.

Algorithm 2 Algoritmo propuesto 1

```
1: procedure MWSA(Grafo, Camiones, Leches)
2:   InicializarLobos() ▷ Utilizando heurística K-GWO
3:   while !Criterio De Termino do
4:     for cada Lobo do
5:       if RandomNumber then ▷ Busqueda local
6:         NuevaSolucion  $\leftarrow$  2OPT(SolucionLoboActual)
7:       else
8:         NuevaSolucion  $\leftarrow$  BorrarGranja(SolucionLoboActual)
9:       if eVFunc(NuevaSolucion) > eVFunc(SolucionLoboActual) then
10:        SolucionLoboActual  $\leftarrow$  NuevaSolucion
11:        if CumpleRestricción(SolucionLoboActual) then
12:          if Mejor < eVFunc(SolucionLoboActual) then
13:            Mejor  $\leftarrow$  eVFunc(SolucionLoboActual)
14:            Camino  $\leftarrow$  SolucionLoboActual
15:        if RandomNumber then ▷ Escape
16:          if RandomNumber then
17:            NuevaSolucion  $\leftarrow$  SwapNodoCamiones(SolucionLoboActual)
18:          else
19:            NuevaSolucion  $\leftarrow$  AgregarNodo(SolucionLoboActual)
20:        if AunNoExisteFactible then ▷ Reparación
21:          if RandomNumber then
22:            NuevaSolucion  $\leftarrow$  SwapNodoCamionesESP(SolucionLoboActual)
23:          else
24:            NuevaSolucion  $\leftarrow$  AgregarNodoESP(SolucionLoboActual)
25:
26:
27:       return Camino
28:
```

El MWSA inicializa una manada de lobos utilizando K-GWO para cada uno, donde cada lobo corresponde a una solución clusterizada. Luego de esto realizará un proceso iterativo hasta que la mejor solución no cambie una cantidad definida de veces, donde realiza búsqueda local realizando 2OPT en el camino de alguno de los camiones de forma aleatoria o elimina un nodo si este se encuentra demasiado lejos de forma que eliminarlo aumentará la función de evaluación. Luego de esto revisa si la nueva solución es mejor que la anterior, si es así se "mueve", es decir, la solución actual será la que fue calculada en la búsqueda local. Luego revisará si esta es factible, este es un punto bastante importante, pues permite movimientos a soluciones no factibles, luego guardará como la mejor solución global si es que esta supera la ya existente. Además de realizar búsqueda local, para evitar el estancamiento en un óptimo local, se realiza con cierta probabilidad un "escape", es decir, se expandirá el espacio de búsqueda, para ello se tienen dos opciones, la primera es realizar un swap entre nodos en el camino de dos camiones y la segunda es eliminar un nodo del camino recorrido de un camión para entregarlo al camino de otro camión. Finalmente, en el caso que no exista aun una solución factible en el grupo de lobos, se realizan opciones de reparación, las cuales son de realizar swap y agregar como en las opciones de escape, pero en este caso se busca siempre mover nodos convenientes para cumplir con la cuota mínima.

Algorithm 3 Algoritmo propuesto 2

```
1: procedure RMWSA(Grafo, Camiones, Leches)
2:   InicializarLobos() ▷ Utilizando heurística K-GWO
3:   while !Criterio De Termino do
4:     for cada Lobo do
5:       if RandomNumber then ▷ Busqueda local
6:         NuevaSolucion  $\leftarrow$  2OPT(SolucionLoboActual)
7:       else
8:         NuevaSolucion  $\leftarrow$  BorrarGranja(SolucionLoboActual)
9:       if eVFunc(NuevaSolucion) > eVFunc(SolucionLoboActual) then
10:        if CumpleRestricción(SolucionLoboActual) then
11:          SolucionLoboActual  $\leftarrow$  NuevaSolucion
12:          if Mejor < eVFunc(SolucionLoboActual) then
13:            Mejor  $\leftarrow$  eVFunc(SolucionLoboActual)
14:            Camino  $\leftarrow$  SolucionLoboActual
15:        if RandomNumber then ▷ Escape
16:          if RandomNumber then
17:            NuevaSolucion  $\leftarrow$  SwapNodoCamiones(SolucionLoboActual)
18:          else
19:            NuevaSolucion  $\leftarrow$  AgregarNodo(SolucionLoboActual)
20:        if NoEsFactible then ▷ Reparación
21:          if RandomNumber then
22:            NuevaSolucion  $\leftarrow$  SwapNodoCamionesESP(SolucionLoboActual)
23:          else
24:            NuevaSolucion  $\leftarrow$  AgregarNodoESP(SolucionLoboActual)
25:
26:
27:       return Camino
28:
```

El RMWSA realiza exactamente las mismas operaciones que MWSA, pero no permite movimientos a soluciones infactibles y además cada vez que la solución de un lobo no es factible, prueba un movimiento para que esta lo sea.

4.4 Parámetros

Los parámetros utilizados por ambos algoritmos corresponden a:

- C: Factor de penalización al exceder capacidad.
- P: Factor de penalización al exceder capacidad.
- N: Factor de penalización al ser distintos tipos de leche.
- NN: Factor de penalización al ser distintos tipos de leche.
- Q: Factor del tamaño del conjunto de lobos.
- T: Criterio de termino.
- SCAP: Probabilidad de escapar.
- S1: Probabilidad de realizar swap.

- R1: Probabilidad de realizar swapESP.
- O: Probabilidad de borrar granja.

Los parámetros utilizados únicamente por MWSA:

- TW: Criterio de reparación.
- AD: Porcentaje de diferencia aceptado entre caminos.

Los parámetros utilizados únicamente por RMWSA:

- TT: Tiempo límite de finalización del algoritmo.

5 Experimentos

Debido a la gran cantidad de parámetros, se realizaron distintas pruebas para determinar buenos valores para el algoritmo.

5.1 MWSA

Se realizaron gráficos de puntos, para las soluciones entregadas para 6 distintos seteos de parámetros para el algoritmo MWSA utilizando 6 semillas distintas, los que se muestran a continuación:

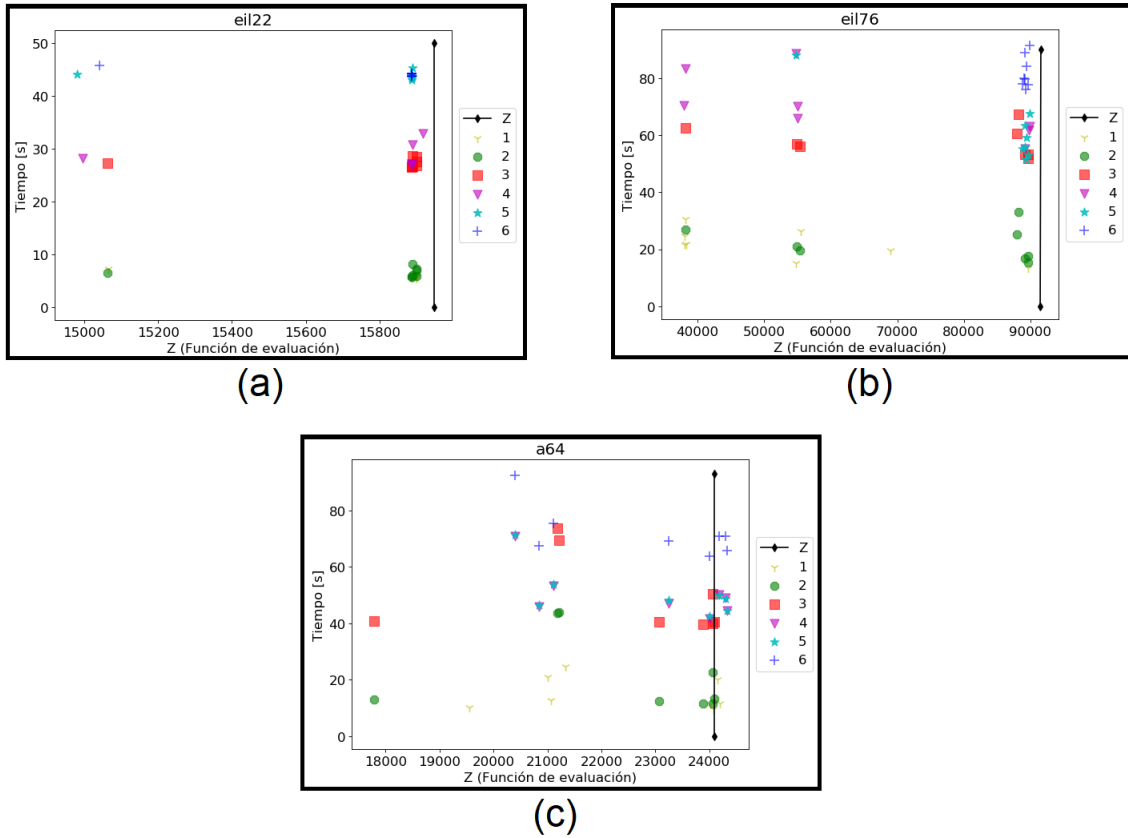


Figura 4: Experimentos realizados para las instancias eil22, eil76 y a64 para MWSA.

Donde 1 corresponde a la cantidad mínima de cada parámetro, 2 al agregar mas cantidad de lobos, 3 al aumentar el criterio de termino, 4 al aumentar la probabilidad de borrar, 5 al aumentar la probabilidad de escapar y 6 a aumentar aun más el criterio de termino. En cada gráfico se observa una linea denominada Z, que corresponde al máximo encontrado en el articulo de Paredes Belmar et al. [11].

La figura (a), corresponde a los resultados para eil22, con 22 nodos, se observa que con una de las semillas, todos los valores obtenidos son cercanos a 15100, mientras que el máximo se encuentra sobre los 15900, mientras que para las otras 5 semillas el valor obtenido es cercano al óptimo, pero en ningún caso llega a serlo.

Para la figura (b), se obtienen resultados de la instancia eil76, es decir, con 76 nodos se observa que para los seteos de parámetros de 5 y 6 se obtienen resultados menos disgregados que los demás.

Finalmente, se observa en (c) que para la instancia a64, los resultados fueron mas variables respecto al tiempo, sobretodo es mas apreciable en el seteo número 2 y 3, donde hay valores mas dispersos en cuanto a tiempo, esto se debe a que el algoritmo fue encontrando mejores soluciones constantemente por lo que el tiempo utilizado para cumplir el criterio de termino fue mas largo.

Al realizar el test no paramétrico de Wilcoxon entre 1 y 3 se obtiene con 98 por ciento de confianza, debido a la cantidad de muestras realizadas debido a un total de 6, para a64:

Wilcoxon Test	Tipo1 -Tipo3
P-Valor	0.4838
Intervalo de confianza	-1587.5 1426.5
Mediana	112.6135

Para esta instancia no es posible asegurar que algún caso sea mejor que otro, el *p-valor* obtenido es poco confiable, por otro lado el intervalo de confianza no se encuentra inclinado para algunos de los valores, mas bien, el limite de uno es cercano al negativo del otro.

Al realizar el test no paramétrico de Wilcoxon entre 1 y 3 se obtiene con 98 por ciento de confianza para eil76,

Wilcoxon Test	Tipo1 -Tipo3
P-Valor	0.04995
Intervalo de confianza	-50703.5 15395.0
Mediana	-24975.15

que no existe evidencia para asegurar que un tipo sea mejor que el otro, si bien el *p-valor* es menor que el caso anterior, el intervalo de confianza muestra una mayor inclinación para el seteo de tipo 3.

Al realizar el test no paramétrico de Wilcoxon entre 1 y 6 se obtiene: con 98 por ciento de confianza para eil76

Wilcoxon Test	Tipo1 -Tipo6
P-Valor	0.01172
Intervalo de confianza	-51366 -10221
Mediana	-38805.5

que el seteo de parámetros del test tipo 1 es mejor que el tipo 6, el *p-valor* posee una magnitud inferior al 0,012 y, a diferencia de los casos anteriores si se puede ver una clara inclinación del intervalo de confianza donde el test del tipo uno predomina sobre el otro.

5.2 RMWSA

Se realizaron gráficos de puntos, para las soluciones entregadas para 10 distintos seteos de parámetros para el algoritmo MWSA utilizando 10 semillas distintas, los que se muestran a continuación:

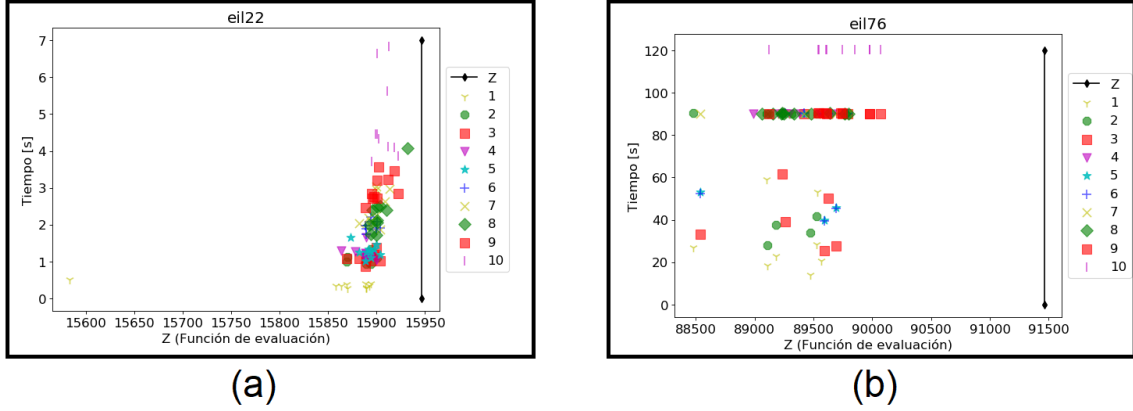


Figura 5: Experimentos realizados para las instancias eil22 y eil76 para RMWSA.

Donde 1 corresponde a la cantidad mínima de cada parámetro, 2 al aumentar el criterio de termino, 3 al aumentar la probabilidad de borrar, 4 corresponde a aumentar la probabilidad de escape, 5 a aumentar el criterio de termino, 6 al agregar mas cantidad de lobos, 7 a disminuir la probabilidad de escape, 8 al aumentar aun mas la cantidad de lobos, 9 a realizar mas iteraciones y 10 a aumentar el tiempo máximo del algoritmo.

La figura (a), corresponde a los resultados para eil22, con 22 nodos, se observa que con una de las semillas, todos los valores obtenidos son cercanos a 15900 y el máximo obtenido por Paredes-Belmar fue de 15950, por lo que en general el algoritmo no se ve con valores dispersos. Además para el seteo 8, se obtiene el valor mas cercano al máximo.

Para la figura (b), se obtienen resultados de la instancia eil76, es decir, con 76 nodos, se observa que para los seteos de parámetros de 9 y 10 se obtienen resultados mas cercanos al máximo, pero no se llega a este en el tiempo máximo de 90 segundos. Para el seteo 10, al aumentar el tiempo máximo de búsqueda a 120 segundos se observa que no existe una mejora de la solución.

Al realizar el test no paramétrico de Wilcoxon entre 2 y 9 se obtiene con 99 por ciento de confianza para eil22:

Wilcoxon Test	Tipo2 -Tipo9
P-Valor	0.02182
Intervalo de confianza	-31.160165 4.031256
Mediana	-14.49168

No es posible asegurar que uno de los dos tipos de seteo de parámetros sea mejor que el otro, el intervalo de confianza presenta valores distribuidos entre los números negativos y positivos.

Al realizar el test paramétrico de Wilcoxon entre 2 y 9 se obtiene con 99 por ciento de confianza para eil76:

Wilcoxon Test	Tipo2 -Tipo9
P-Valor	0.00691
Intervalo de confianza	-882.226594 -6.093722
Mediana	-502.3906

Si es posible asegurar que el set de parámetros de tipo 2 es mejor que el tipo 9, todos los valores del intervalo de confianza se encuentran inclinados para el seteo de tipo 2.

6 Resultados

Definiendo el seteo de parámetros como el numero 3 visto en la sección anterior, se realizaron 10 pruebas con distintas semillas para cada una de las instancias disponibles, las cuales se muestran en la siguiente tabla:

Instancia	Z Obtenido	Desviación Z	Tiempo	Z Paper	Tiempo Paper
a36	28821	104	37	29233	110
a62	20158	1417	46	22917	1022
a64	22422	2100	49	24100	5395
a65	19839	7293	54	28046	478
eil22	15773	290	27	15947	12
eil51	25594	4468	53	50128	154
eil76	74154	19744	58	91461	1700
tai75A	35116	100	50	65477	13760

Table 1: Tabla de valores obtenidos para MWSA.

Donde cada valor, es el promedio de las 10 mediciones. Se observa que en cuanto a tiempo el algoritmo MWSA, utiliza un tiempo bastante reducido en comparación al Paper en la mayoría de los casos, encontrando siempre soluciones de peor calidad, pero que en las instancias a36, a62, a64 y eil22 se acerca bastante al óptimo en poco tiempo, mientras que en las restantes los valores obtenidos no son tan cercanos.

Definiendo el seteo de parámetros como el numero 3 visto en la sección anterior, se realizaron 10 pruebas con distintas semillas para eil22 y eil76, debido a que el algoritmo no lograba encontrar soluciones factibles en un tiempo corto para las de tipo a y tipo tai. Esto se debe a que los movimientos realizados pueden estar mal implementados y realizar que un camión sobrepase su capacidad, por lo que la solución no sera factible y al no poder moverse a soluciones infactibles realizar otros movimientos para salir del estancamiento no fue posible. La tabla obtenida se muestra a continuación:

Instancia	Z Obtenido	Desviación Z	Tiempo	Z Paper	Tiempo Paper
eil22	15903	10	3	15947	12
eil76	89691	263	90	91461	1700

Table 2: Tabla de valores obtenidos para RMWSA.

Donde cada valor, es el promedio de las 10 mediciones. Se observa que el tiempo del algoritmo es increíblemente reducido en comparación al Paper, e incluso llega a soluciones bastante cercanas al óptimo.

A continuación se muestran los resultados obtenidos del MWSA, en comparación al tiempo del artículo:

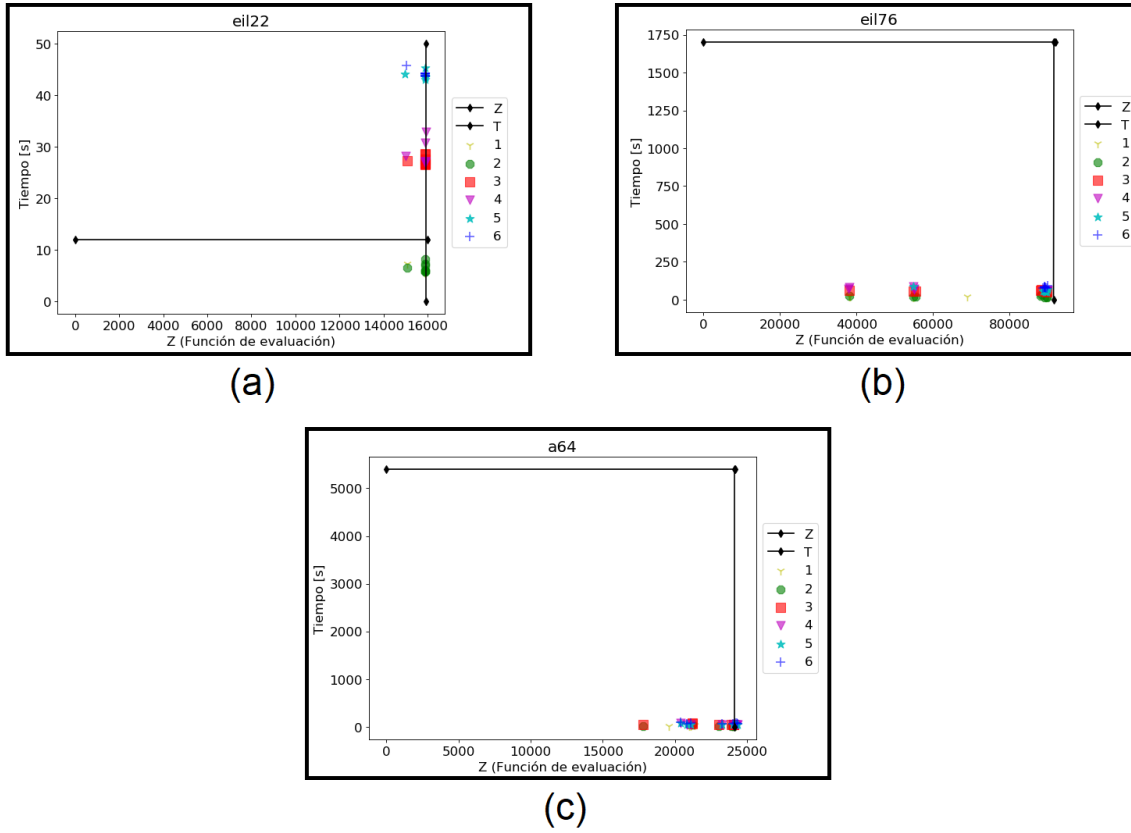


Figura 6: Comparación temporal para las instancias eil22, eil76 y a64 para RMWSA.

Se observa que en (a) para 22 nodos el tiempo es mayor y el resultado se acerca bastante el óptimo sin tener una desviación muy grande, mientras que en (b) para 76 nodos el tiempo es bastante reducido pero obtiene solo en algunos casos soluciones cercanas al óptimo. Finalmente en (c) para 64 nodos, los resultados obtenidos no están tan dispersos y muestran un tiempo significante mente menor.

Los siguientes gráficos muestran los resultados obtenidos del RMWSA, en comparación al tiempo del artículo:

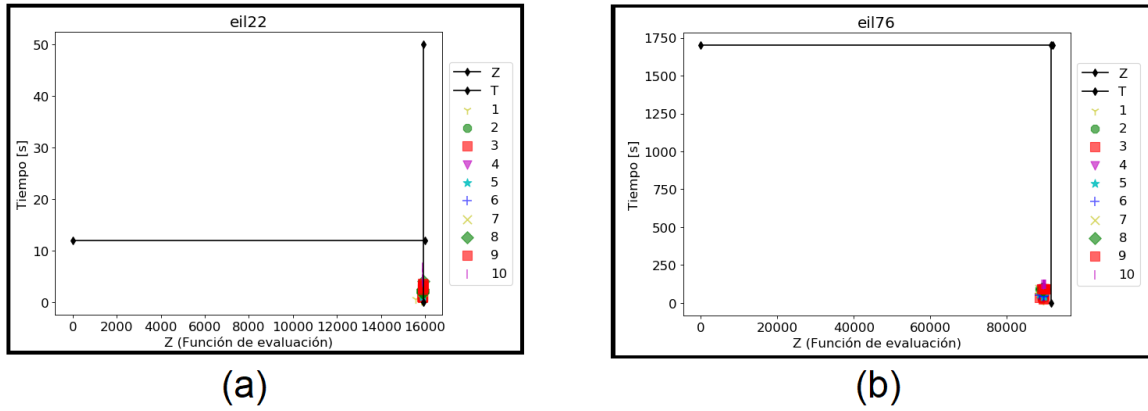


Figura 7: Comparación temporal para las instancias eil22 y eil76 para RMWSA.

Se observa que tanto para (a) como (b) el resultado es muchísimo mas rápido que el algoritmo utilizado por Paredes-Belmar et. al. y se acerca bastante al óptimo.

Al realizar el test no paramétrico de Wilcoxon entre MWSA y RMWSA se obtiene con 98 por ciento de confianza, debido a tener 6 semillas de prueba para eil22:

Wilcoxon Test	Valores MWSA - RMWSA
P-Valor	0.01729
Intervalo de confianza	-430.5630492 -0.1859921
Mediana	-14.29021

El algoritmo MWSA es mejor que el algoritmo RMWSA, el intervalo de confianza presenta valores únicamente en los negativos.

En tiempo:

Wilcoxon Test	Tiempos MWSA - RMWSA
P-Valor	0.01172
Intervalo de confianza	23.29281 25.37839
Mediana	24.2712

El algoritmo RMWSA presenta un mejor desempeño en tiempo de ejecución, con un *p-valor* con una magnitud no mayor que 0,012 y un intervalo claramente inclinado hacia los números positivos.

Al realizar el test no paramétrico de Wilcoxon entre MWSA y RMWSA se obtiene con 98 por ciento de confianza para eil76:

Wilcoxon Test	Valores MWSA - RMWSA
P-Valor	0.01729
Intervalo de confianza	-43257.86328 -69.75385
Mediana	-17510.64

Nuevamente, es posible asegurar un mejor desempeño del algoritmo MWSA, no existen elementos en el intervalo de confianza que demuestren lo contrario.

En tiempo:

Wilcoxon Test	Tiempos MWSA - RMWSA
P-Valor	0.01172
Intervalo de confianza	-37.81245 -25.28408
Mediana	-33.0436

Para este caso, a diferencia del anterior, el algoritmo MWSA presenta un mejor tiempo de ejecución que el algoritmo RMWSA.

7 Conclusiones

En este informe se realiza un análisis y propuesta de solución para el MCwB. Existe una motivación en la resolución de este problema debido a lo beneficioso de realizar mejores rutas en las empresas productoras de leche, las cuales hacen parte de este proceso constantemente.

El Milk collection es un problema muy parecido a un VRP pero con una inmensa cantidad de restricciones distintas dependiendo de la realidad a la que será aplicado lo que lo hace más complejo y con distintas formas de plantear pero puede ser muy beneficioso para las empresas productoras de leche debido a lo complejo del problema al obtener buenas soluciones. Debido a lo complejo del problema asociado MCwB al ser NP-Difícil se han utilizado algoritmos reparadores, Ant colony System y algoritmos evolutivos para resolverlo. En los casos estudiados por Paredes-Belmar et al. [11] [10] a pesar de encontrar buenas soluciones, ambos algoritmos utilizan un tiempo extenso para grandes cantidades de nodos, además se plantea la heurística HP, la cual mejora los resultados en ambos sentidos. Todas las soluciones realizadas hasta ahora son similares en el uso de grafos y técnicas de resolución para VRP, junto con adecuaciones específicas para el problema.

Se proponen dos algoritmos que utilizan adecuaciones de algoritmos basados en la naturaleza, principalmente en el comportamiento de los lobos. Estos algoritmos son bastante rápidos, pero ninguno de los dos logra superar por ahora las soluciones del ACO utilizado por Paredes-Belmar et al. [11].

Se espera mejorar el algoritmo RMWSA, pues su problema está relacionado a los movimientos de reparación realizados. Encontrando mejores algoritmos reparadores e implementándolos al algoritmo, podrá entregar mejores soluciones para todo tipo de grafo. Por lo que se espera a futuro que este algoritmo logre competir con otros algoritmos en cualquier tipo de VRP con distintas restricciones.

8 Bibliografía

- [1]
- [2] Martin Butler, Pat Herlihy, and Peter B. Keenan. Integrating information technology and operational research in the management of milk collection. *Journal of Food Engineering*, 70(3):341 – 349, 2005. Operational Research and Food Logistics.
- [3] Gendreau M. Laporte Cordeau, J.-F. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30:105 – 119, 1997.

- [4] George Dantzig and John Ramser. The truck dispatching problem. *Management Science*, 6:80–91, 1959.
- [5] C. Duhamel, P. Lacomme, A. Quilliot, and H. Toussaint. 2l-cvrp: A grasp resolution scheme based on rcsp. In *2009 International Conference on Computers Industrial Engineering*, pages 1094–1099, July 2009.
- [6] GÃneÃŸ ErdoÃŸan. An open source spreadsheet solver for vehicle routing problems. *Computers Operations Research*, 84:62 – 72, 2017.
- [7] Luis Miguel Escobar Falc3n, David Alvarez Martinez, Mauricio Granada-Echeverri, John Escobar, and Rub3n Augusto Romero-L3zaro. A matheuristic algorithm for the three-dimensional loading capacitated vehicle routing problem (3l-cvrp). 2016, 03 2015.
- [8] M. Khorsid L. Korayem and S. S. Kassem. Using grey wolf algorithm to solve the capacitated vehicle routing problem. *IOP Conference Series: Materials Science and Engineering*, 83, 2015.
- [9] Renaud Masson, Nadia Lahrichi, and Louis-Martin Rousseau. A two-stage solution method for the annual dairy transportation problem. *European Journal of Operational Research*, 251(1):36 – 43, 2016.
- [10] Germ3n Paredes-Belmar, Armin L3er-Villagra, Vladimir Marianov, Cristi3n E. Cort3s, and Andr3s Bronfman. The milk collection problem with blending and collection points. *Computers and Electronics in Agriculture*, 134:109 – 123, 2017.
- [11] Germ3n Paredes-Belmar, Vladimir Marianov, Andr3s Bronfman, Carlos Obreque, and Armin L3er-Villagra. A milk collection problem with blending. *Transportation Research Part E: Logistics and Transportation Review*, 94:26 – 43, 2016.
- [12] R.J Petch and S Salhi. A multi-phase constructive heuristic for the vehicle routing problem with multiple trips. *Discrete Applied Mathematics*, 133(1):69 – 92, 2003. International Symposium on Combinatorial Optimisation.
- [13] S. Fong Q. Song and R. Tang. Self-adaptive wolf search algorithm. *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, pages 576–582, 2016.
- [14] X. Yang R. Tang, S. Fong and S. Deb. Wolf search algorithm with ephemeral memory. *Seventh International Conference on Digital Information Management (ICDIM 2012)*, pages 165–172, Macau, 2012.
- [15] Kanchana Sethanan and Rapeepan Pitakaso. Differential evolution algorithms for scheduling raw milk transportation. *Computers and Electronics in Agriculture*, 121:245 – 259, 2016.
- [16] Boon Ean Teoh, S. G. Ponnambalam, and G. Kanagaraj. Differential evolution algorithm with local search for capacitated vehicle routing problem. *Int. J. Bio-Inspired Comput.*, 7(5):321–342, October 2015.
- [17] Dilek Tuzun, Michael A. Magent, and Laura I. Burke. Selection of vehicle routing heuristic using neural networks. *International Transactions in Operational Research*, 4(3):211 – 221, 1997.
- [18] Alex Van Breedam. Improvement heuristics for the vehicle routing problem based on simulated annealing. *European journal of operational research*, 86, 480-490. 86(3):480–490, 02 1995.

- [19] Y. Zhang, J. Liu, F. Duan, and J. Ren. Genetic algorithm in vehicle routing problem. In *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)*, volume 2, pages 578–581, Nov 2007.